Logos 系列 FPGA 专用 RAM 模块（DRM）用户指南

(UG020002, Version1.1)

(2018.12.18)

深圳市紫光同创电子有限公司

文档版本修订记录

版本号	发布日期	修订记录
V1.0	2018/4/6	初始版本
V1.1	2018/12/18	1. 封面增加最新的版本更新时间；页脚去掉版本信息； 2. DRM 支持的原语 GTP_DRM18K_E1 名称改为 GTP_DRM18K

名词术语解释

DRM：Dedicated RAM Module

SP：Single Port

SDP：Simple Dual Port

NW：Normal-Write

RBW：Read-before-Write

TW：Transparent-Write

OR：Output Register

目录

图目录

表目录

关于本手册

本手册为Logos系列FPGA产品的DRM （Dedicated RAM Module，专用RAM模块）用户指南文档。DRM为Logos系列产品提供了单个18Kbits或者两个9Kbits的存储空间。DRM支持多种工作模式：DP RAM模式，SDP RAM模式，SP RAM模式和ROM模式以及同步\异步FIFO模式。

一、概述

Logos系列FPGA的DRM有高达18K bits的存储单元并且容量可被独立配置为2个9K bits或者1个18K bits。每个DRM都能支持DP（True Dual Port，双口）RAM模式，同时也以被配置为SP（Single Port，单口）RAM模式， SDP（Simple Dual Port，简单双口）RAM模式，ROM模式，以及可选丢包重发的同步\异步FIFO模式。DRM资源还支持输入寄存器（IR）、输出寄存器（OR）以及Core Latch，这使得DRM级联使用时拥有更加出色的性能表现。DRM的总数取决于Logos系列器件类型。

DP RAM的端口数据位宽高至18bit，它的两个端口除了共享RAM内容之外完全独立，并支持不同的时钟域；SDP RAM的端口数据位宽高至36bit，它的两个端口也支持不同的时钟域，但它的一个端口仅限于写操作，另一个端口仅限于读操作。使用少量的外部CLM资源，Logos系列的SDP RAM还可以扩展成FIFO。ROM模式下，ROM的端口数据位宽高至36bits。DRM的内容通常在下载配置数据的过程中初始化。当然，其它模式也可以利用编程配置来初始化DRM的内容。在同步或者异步FIFO模式时，一个端口专用于FIFO数据写入，另一个端口专用于FIFO数据读取，读写端口可以支持不同的时钟，但不支持混合位宽及Byte enable的写操作。FIFO模式在同步和异步模式下都支持可选的丢包重发模式。如表1所示为DRM功能特性列表。

嵌入的DP RAM、SP RAM、SDP RAM、ROM以及可选丢包重发的同步\异步FIFO模块，都能便捷地通过深圳市紫光同创电子有限公司的软件Pango Design Suite内嵌的IP Compiler工具生成。

（一）功能特性列表

如表 1 所示，为 Logos 系列 FPGA 的 DRM 功能特性列表。

表 1 功能特性列表

特性列表	
存储容量	18K(或 2 个 9K)
输入寄存器	有
输出寄存器	有（可旁路）
最大数据总线宽度	36 位
SP RAM 模式	支持最大 36 位数据宽度
DP RAM 模式	支持最大 18 位数据宽度
SDP RAM 模式	支持最大 36 位数据宽度
Byte 写使能	支持
Parity	支持单位字节附加一位信息存储位
Transparent 写模式	支持
Read-before-Write 写模式	支持
地址输入保持	支持
端口独立输出时钟极性控制	支持
内置 FIFO 功能	支持
同步/异步 FIFO 控制	支持
FIFO 写入端的丢包功能	支持
FIFO 读出端的重发功能	支持
FIFO 级联	支持
可编程的 ALMOST_FULL/ALMOST_EMPTY 指示	支持
FIFO 容量	18K

（二）支持的配置

DP RAM和SDP RAM均支持9Kb模式和18Kb模式。在DP RAM模式下，DRM有18-bit的数据最大位宽，DRM模块中A、B两个端口均可以独立进行读写操作，且均支持不同的时钟。而SDP RAM模式下，DRM数据位宽则增大至36 bits，A， B两个端口中一个端口专用于数据写入，另一个端口专用于数据读取，读写端口同样均支持不同的时钟。

在DP RAM模式下，两个端口的位宽可以独立地根据DRM资源进行配置。9 Kb DRM模式下的DP RAM配置，SDP RAM配置，以及18 Kb DRM模式下的DP RAM配置，SDP RAM配置已在表2，表3，表5和表6描述。

在SP RAM和ROM两种模式的9Kb模式下，DRM包含两个端口，SP RAM模式可以分别对这两个端口独立进行读写操作，而ROM模式下该端口只读；在两对端口共享情况下，DRM仅包含一个端口，

SP RAM模式可以对该端口进行读写操作，而ROM模式下该端口只读。9 Kb模式下的SP RAM模式和ROM模式列表，以及18 Kb模式下的SP RAM模式和ROM模式列表已在表4和表7描述。

FIFO模式下，一个端口专用于数据写入另一个端口专用于数据读取，读写端口可以采用不同的时钟。

地址和数据端口Mapping详见附录。

1） 9 Kb DRM——True Dual Port RAM 模式

如表 2 所示为 9Kb DRM 模式下 True Dual Port RAM 模式允许的位宽组合。

表 2 9Kb DRM 模式下 True Dual Port RAM 模式允许的位宽组合

<table>
<tr><td colspan="3" rowspan="3"></td><td colspan="7">B 端口 0/1</td></tr>
<tr><td colspan="5">No Parity Bits</td><td colspan="2">With Parity Bits</td></tr>
<tr><td>8Kx1</td><td>4Kx2</td><td>2Kx4</td><td>1Kx8</td><td>512x16</td><td>1Kx9</td><td>512x18</td></tr>
<tr><td rowspan="7">A 端口 0/1</td><td rowspan="5">No Parity Bits</td><td>8Kx1</td><td colspan="5" rowspan="5">DP RAM 可用</td><td colspan="2" rowspan="5">None Allowed</td></tr>
<tr><td>4Kx2</td></tr>
<tr><td>2Kx4</td></tr>
<tr><td>1Kx8</td></tr>
<tr><td>512x16</td></tr>
<tr><td rowspan="2">With Parity Bits</td><td>1Kx9</td><td colspan="5" rowspan="2">None Allowed</td><td colspan="2" rowspan="2">DP RAM 可用</td></tr>
<tr><td>512x18</td></tr>
</table>

2） 9 Kb DRM——Simple Dual Port RAM 模式

如表 3 所示为 9Kb DRM 模式下 Simple Dual Port RAM 模式允许的位宽组合。

表 3 9Kb DRM 模式下 Simple Dual Port RAM 模式允许的位宽组合

<table>
<tr><td colspan="3" rowspan="3"></td><td colspan="9">写端口 0/1</td></tr>
<tr><td colspan="6">No Parity Bits</td><td colspan="3">With Parity Bits</td></tr>
<tr><td>8Kx1</td><td>4Kx2</td><td>2Kx4</td><td>1Kx8</td><td>512x16</td><td>256x32</td><td>1Kx9</td><td>512x18</td><td>256x36</td></tr>
<tr><td rowspan="9">读端口 0/1</td><td rowspan="6">No Parity Bits</td><td>8Kx1</td><td colspan="6" rowspan="6">SDP RAM 可用</td><td colspan="3" rowspan="6">None Allowed</td></tr>
<tr><td>4Kx2</td></tr>
<tr><td>2Kx4</td></tr>
<tr><td>1Kx8</td></tr>
<tr><td>512x16</td></tr>
<tr><td>256x32</td></tr>
<tr><td rowspan="3">With Parity Bits</td><td>1Kx9</td><td colspan="6" rowspan="3">None Allowed</td><td colspan="3" rowspan="3">SDP RAM 可用</td></tr>
<tr><td>512x18</td></tr>
<tr><td>256x36</td></tr>
</table>

3） 9 Kb DRM——Single Port RAM 模式和 ROM 模式

如表 4 所示为 9Kb DRM 模式下 Simple Port RAM 模式和 ROM 模式列表。

表 4 9Kb DRM 模式 Single Port RAM 模式和 ROM 模式列表

<table>
<tr><td colspan="2"></td><td>8K*1</td><td>4K*2</td><td>2K*4</td><td>1K*8</td><td>512*16</td><td>256*32</td><td>1K*9</td><td>512*18</td><td>256*36</td></tr>
<tr><td rowspan="2">模式</td><td>SP RAM</td><td colspan="9">SP RAM 可用</td></tr>
<tr><td>ROM</td><td colspan="9">ROM 可用</td></tr>
</table>

4） 18 Kb DRM——True Dual Port RAM 模式

如表 5 所示为 18Kb DRM 模式下 True Dual Port RAM 模式允许的位宽组合。

表 5 18Kb DRM 模式下 True Dual Port RAM 模式允许的位宽组合

<table>
<tr><td colspan="3" rowspan="3"></td><td colspan="7">B 端口 0/1</td></tr>
<tr><td colspan="5">No Parity Bits</td><td colspan="2">With Parity Bits</td></tr>
<tr><td>16Kx1</td><td>8Kx2</td><td>4Kx4</td><td>2Kx8</td><td>1Kx16</td><td>2Kx9</td><td>1Kx18</td></tr>
<tr><td rowspan="7">A 端口 0/1</td><td rowspan="5">No Parity Bits</td><td>16Kx1</td><td colspan="5" rowspan="5">DP RAM 可用</td><td colspan="2" rowspan="5">None Allowed</td></tr>
<tr><td>8Kx2</td></tr>
<tr><td>4Kx4</td></tr>
<tr><td>2Kx8</td></tr>
<tr><td>1Kx16</td></tr>
<tr><td rowspan="2">With Parity Bits</td><td>2Kx9</td><td colspan="5" rowspan="2">None Allowed</td><td colspan="2" rowspan="2">DP RAM 可用</td></tr>
<tr><td>1Kx18</td></tr>
</table>

5） 18 Kb DRM——Simple Dual Port RAM 模式

如表 6 所示为 18Kb DRM 模式下 Simple Dual Port RAM 模式允许的位宽组合。

表 6 18Kb DRM 模式下 Simple Dual Port RAM 模式允许的位宽组合

<table>
<tr><td colspan="3" rowspan="3"></td><td colspan="9">写端口 0/1</td></tr>
<tr><td colspan="6">No Parity Bits</td><td colspan="3">With Parity Bits</td></tr>
<tr><td>16Kx1</td><td>8Kx2</td><td>4Kx4</td><td>2Kx8</td><td>1Kx16</td><td>512x32</td><td>1Kx9</td><td>512x18</td><td>256x36</td></tr>
<tr><td rowspan="9">读端口 0/1</td><td rowspan="6">No Parity Bits</td><td>8Kx1</td><td colspan="6" rowspan="6">SDP RAM 可用</td><td colspan="3" rowspan="6">None Allowed</td></tr>
<tr><td>4Kx2</td></tr>
<tr><td>2Kx4</td></tr>
<tr><td>1Kx8</td></tr>
<tr><td>512x16</td></tr>
<tr><td>256x32</td></tr>
<tr><td rowspan="3">With Parity Bits</td><td>2Kx9</td><td colspan="6" rowspan="3">None Allowed</td><td colspan="3" rowspan="3">SDP RAM 可用</td></tr>
<tr><td>1Kx18</td></tr>
<tr><td>512x36</td></tr>
</table>

6） 18 Kb DRM——Single Port RAM 模式和 ROM 模式

如表 5 所示为 9Kb DRM 模式下 Simple Port RAM 模式和 ROM 模式列表。

表 7 9Kb DRM 模式 Single Port RAM 模式和 ROM 模式列表

<table>
<tr><td colspan="2"></td><td>16K*1</td><td>8K*2</td><td>4K*4</td><td>2K*8</td><td>1K*16</td><td>512*32</td><td>2K*9</td><td>1K*18</td><td>512*36</td></tr>
<tr><td rowspan="2">模式</td><td>SP RAM</td><td colspan="9">SP RAM 可用</td></tr>
<tr><td>ROM</td><td colspan="9">ROM 可用</td></tr>
</table>

（三） Logos 系列 FPGA DRM 使用规则

DP RAM和SDP RAM都有两个相对独立的端口，若同时通过两个端口对同一地址进行读写操作会引起冲突。同地址写写、读写冲突在DRM中属于非法操作，其当前操作不能正常完成，所以需要在实际应用中通过用户逻辑加以规避。

二、DRM 原语

Logos 系列 FPGA DRM 的原语是各种模式的基础，其原语有两种：GTP_DRM9K 和 GTP_DRM18K，其支持类型如表 8 所示：

表 8 Logos 系列 FPGA DRM 原语

原语	支持类型
GTP_DRM9K	支持 x1,x2，x4，x8，x16，x32（和 x9，x18，x36）的数据位宽
GTP_DRM18K	支持 x1,x2，x4，x8，x16，x32（和 x9，x18，x36）的数据位宽

注:1. 所有原语模块在软件安装路径..\arch\vendor\pango\verilog\simulation可找出。

三、DP RAM 和 SDP RAM

（一）数据端口

DP RAM 模式下，18Kb、9Kb 的 DP RAM 拥有除了共享 DRM 内容之外完全独立的两个端口：A 端口和 B 端口。它们的结构完全对称，如图 1 所示， 表 9 罗列了端口名称和端口的描述。

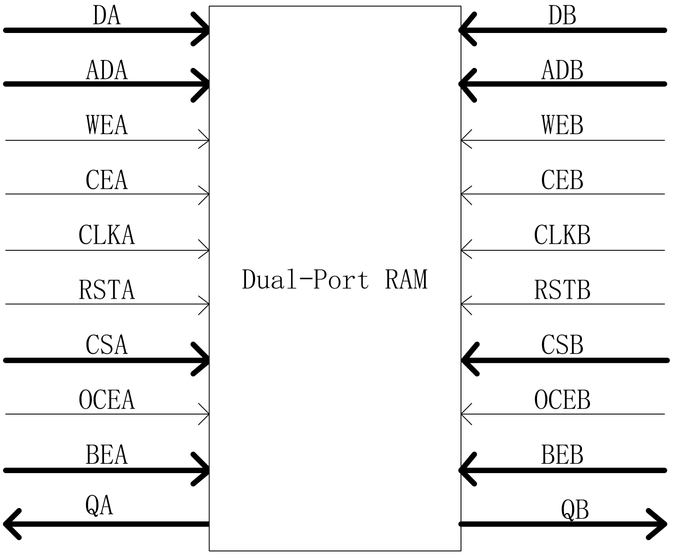


图 1 DP RAM 数据端口

表 9 DP RAM 端口命名及描述

端口命名	输入/输出	描述	端口命名	输入输出	描述
DA	输入	A 端口数据输入	DB	输入	B 端口数据输入
ADA	输入	A 端口地址输入	ADB	输入	B 端口地址输入
WEA	输入	A 端口写使能	WEB	输入	B 端口写使能
CEA	输入	A 端口时钟使能	CEB	输入	B 端口时钟使能
CLKA	输入	A 端口时钟	CLKB	输入	B 端口时钟
RSTA	输入	A 端口输出寄存器复位	RSTB	输入	B 端口输出寄存器复位
CSA	输入	A 端口地址扩展	CSB	输入	B 端口地址扩展
OCEA	输入	A 端口读出寄存器 OE	OCEB	输入	B 端口读出寄存器 OE
BEA	输入	A 端口字节使能控制	BEB	输入	B 端口字节使能控制
QA	输出	A 端口数据输出	QB	输出	B 端口数据输出

注:1. 具体数据位宽详见附录字节附加信息位。

（二）读操作

DP RAM 支持：

- DP 读操作
- DP 写操作
- 一端口读一端口写操作
- 两端口有独立的位宽设置

SDP RAM 模式支持：

- 双端口读和写，但是其中一个端口作为读端口，另一个端口作为写端口。
- 任意一个端口不能同时做读或者写操作。
- 两端口有独立的位宽设置
- x32/x36 位宽 A 端口固定为写入端口（WEA[0/1]为高时（default 极性）有效），B 端口固定为读出端口（WEB[0/1]为低时（default 极性）有效）。

在DRM单元中，A/B两个端口都包括有数据输出Latch。在读操作中，输出寄存器（OR）旁路时，QA/QB输出为Latch输出，在同一个读时钟周期时钟上沿（默认时钟极性）输出。

（三）写操作模式

根据数据写入时同一端口输出的数据的不同，DRM 的端口写操作支持 Normal Write mode（NW），Transparent Write mode（TW），Read before Write mode（RBW）三种模式。如图 2 为三种写操作模式的时序图。

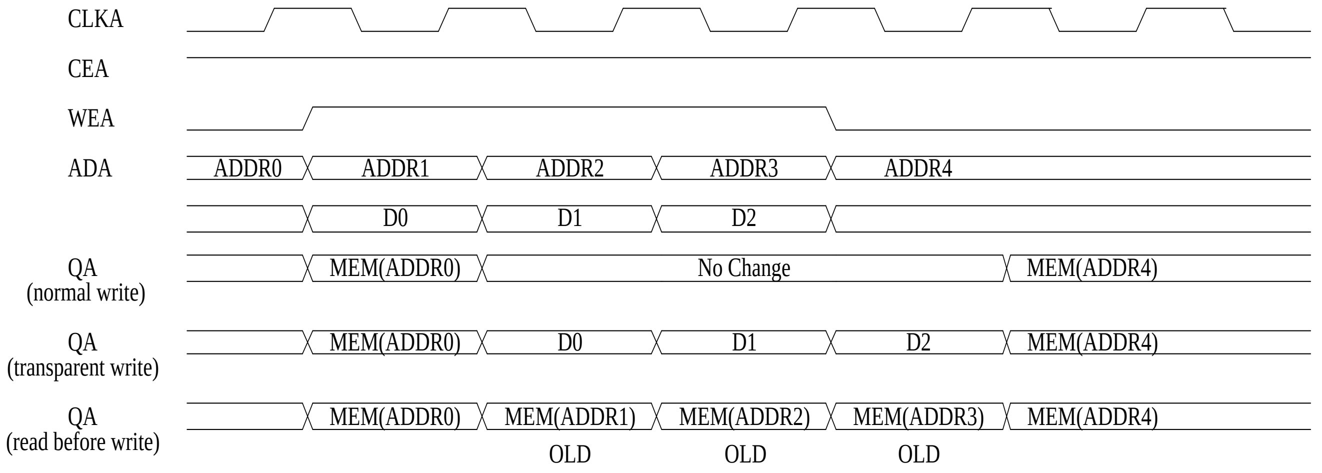


图 2 三种写操作模式输出时序图

注：1.DRM 的端口写操作模式仅适用于具有读写功能的端口，即 DP RAM 的 A/B 端口与 SP RAM 的端口。

2.Default 为 NW 写操作模式。

1） Normal Write mode

如图 2 三种写操作模式输出时序图的 normal write 或如图 3 Normal Write mode 读写时序图，当用户从 DRM 的一个端口写入数据时，此时该端口的输出数据不更新。

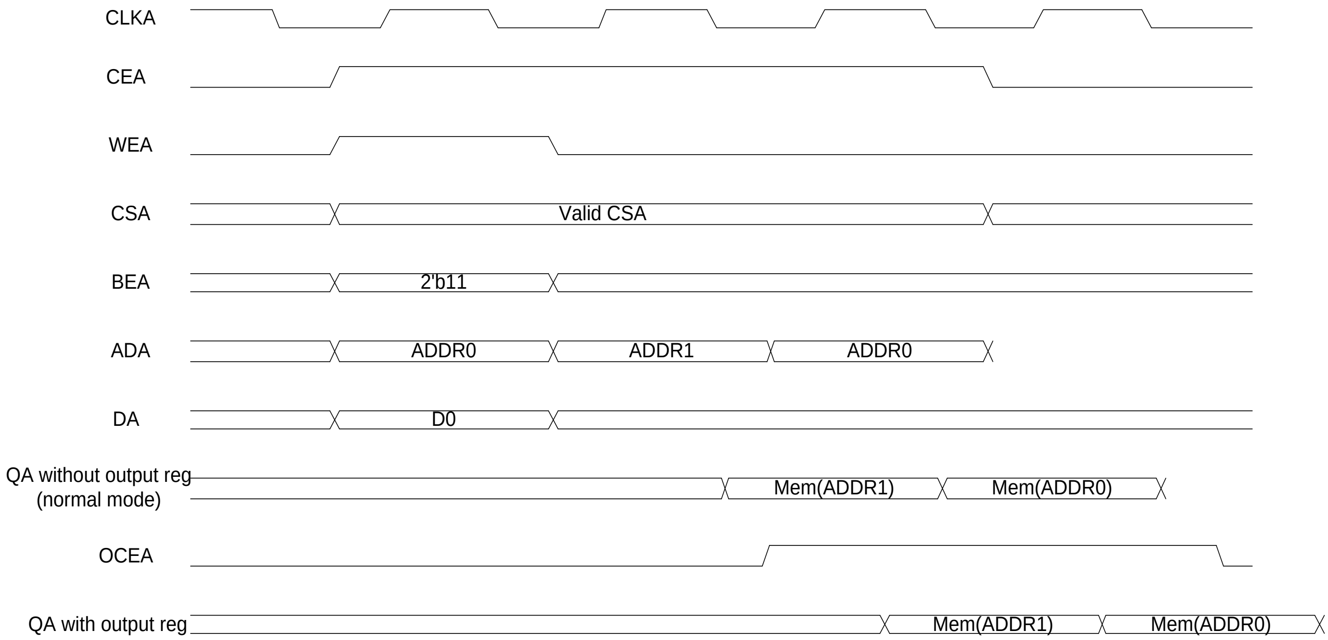


图 3 Normal Write mode 读写时序图

2） Transparent Write mode

图 2 三种写操作模式输出时序图的 transparent write 或如图 4 Transparent Write mode 读写时序图，当用户从 DRM 的一个端口写入数据时，写入数据在写入 RAM 的同时（即写操作的下一个时钟周期）直接输出到输出端口。

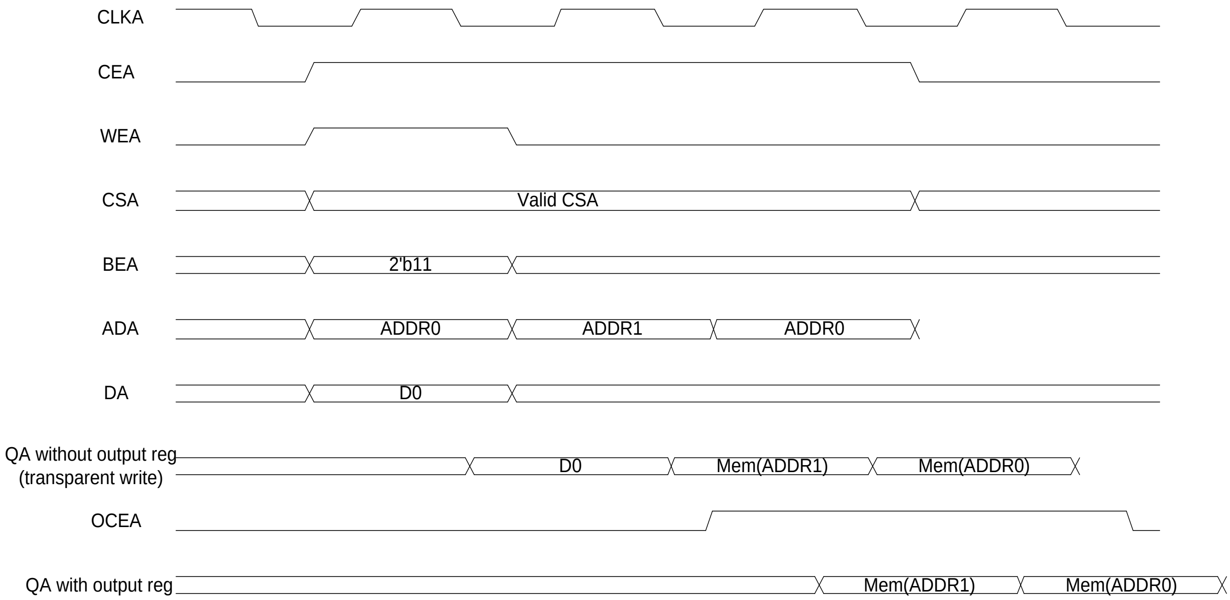


图 4 Transparent Write mode 读写时序图

3） Read before Write mode

如图 2 三种写操作模式输出时序图的 read before write 或如图 5 Read before Write mode 读写时序图，当用户从 DRM 的一个端口写入数据时，将首先读出该地址所引的原数据并在写操作的下一个时钟周期输出到输出端口。

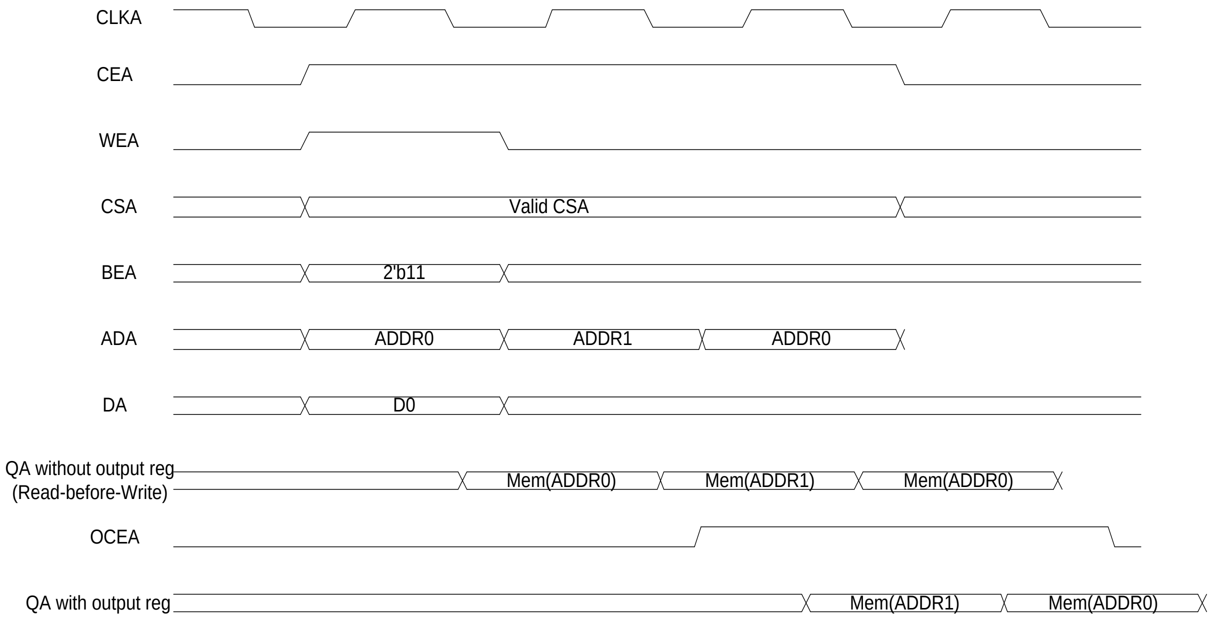


图 5 Read before Write mode 读写时序图

（四）Byte Write 模式

DRM 支持写操作的 Byte-Write 模式，通过 BE 信号（高有效）实现对选定数据字节写入，同时屏蔽对同一地址索引的其它字节的写入。该模式主要应用于在限定数据总线宽度时，只对比较窄的数据总线进行操作。比如可以在 18 位数据总线上只对 9 位宽度数据操作。

（五） Byte Enable

当端口位宽为2^N bit，支持 16/32 bit 的写操作的 Byte Enable，此时每个字节包含 8 bit。当端口位宽为9×2^N bit 时，支持 18/36 bit 的写操作的 Byte Enable，此时每个字节包含 9 bit。例如，在端口数据位宽为 32 bit 时，BE 包含四个比特，BE[3]控制 data[31:24]；BE[2]控制 data[23:16]；依次类推。当 BE[3:0]=4’b0001 时，写操作有效时仅 data[7:0]被写入当前地址。设计时有以下三点需要考虑的地方：

- A 端口和 B 端口分别有独立的字节使能控制
  A0 端口：BEA0[1:0]，B0 端口：BEB0[1:0]
  A1 端口：BEA1[1:0]，B1 端口：BEB1[1:0]。
- 字节使能信号需要在 x18 位数据宽度模式下总是有效，当不使用时，需要在 PDS 软件将该信号连接到“1”。
- 字节使能信号复用低位地址信号：
  BEA0[1:0] =ADA0[1:0]，
  BEB0[1:0] =ADB0[1:0]（x32/x36 数据位宽模式下，ADA0[3:2]）。
  BEA1[1:0] =ADA1[1:0]，
  BEB1[1:0] =ADB1[1:0]（x32/x36 数据位宽模式下，ADA1[3:2]）。

表 10 罗列了 Byte Enable 信号对写入数据位的控制

表 10 Byte Enable 信号控制

DRM 数据总线宽度	字节宽度	Byte EnBble 信号	实际写入数据位
18 位	18 位	BEA0[1:0]=2'b11	DA0[17:0]
		BEB0[1:0]=2'b11	DB0[17:0]
		BEA1[1:0]=2'b11	DA1[17:0]
		BEB1[1:0]=2'b11	DB1[17:0]
	9 位	BEA0[1:0]=2'b01	DA0[8:0]
		BEA0[1:0]=2'b10	DA0[17:9]
		BEB0[1:0]=2'b01	DB0[8:0]
		BEB0[1:0]=2'b10	DB0[17:9]
		BEA1[1:0]=2'b01	DA1[8:0]
		BEA1[1:0]=2'b10	DA1[17:9]
		BEB1[1:0]=2'b01	DB1[8:0]
		BEB1[1:0]=2'b10	DB1[17:9]
36 位	36 位	{BEB0,BEA0}=4'b1111	{DB0[17:0],DA0[17:0]}
		{BEB1,BEA1}=4'b1111	{DB1[17:0],DA1[17:0]}
	18 位	{BEB0,BEA0}=4'b0011	DA0[17:0]
		{BEB0,BEA0}=4'b1100	DB0[17:0]
		{BEB1,BEA1}=4'b0011	DA1[17:0]
		{BEB1,BEA1}=4'b1100	DB1[17:0]
	9 位	{BEB0,BEA0}=4'b0001	DA0[8:0]
		{BEB0,BEA0}=4'b0010	DA0[15:9]
		{BEB0,BEA0}=4'b0100	DB0[8:0]
		{BEB0,BEA0}=4'b1000	DB0[17:9]
		{BEB1,BEA1}=4'b0001	DA1[8:0]
		{BEB1,BEA1}=4'b0010	DA1[15:9]
		{BEB1,BEA1}=4'b0100	DB1[8:0]
		{BEB1,BEA1}=4'b1000	DB1[17:9]

注：1.BW 写操作模式支持 x18（x16）、x36（x32）数据端口宽度

2.BW 写操作模式可以与 NW、TW 或 RBW 写操作模式共存

（六）端口地址冲突仲裁

DP RAM 和 SDP RAM 都有两个相对独立的端口，若同时通过两个端口对同一地址进行读写操作会引起冲突。同地址写写、读写冲突在 DRM 中属于非法操作，其当前操作不能正常完成，所以需要在实际应用中通过用户逻辑加以规避。

（七）可选的输出寄存器

针对数据输出端口，DRM 特别提供了可选的输出寄存器，以取得更好的时序性能。如图 6 所示，输出寄存器可由独立的使能信号 OCE 控制，当 OCE 为常量 1 时，输出寄存器的使用会把读操作的延迟从一个时钟周期增加到两个时钟周期（Case 1）；在具有流控功能的流水线设计应用中，用户也可以用逻辑灵活控制 OCE（Case 2）。



图 6 带输出寄存器的读时序图

（八） SDP RAM

每个 9Kb、18Kb 的 DRM 也能被配置为 SDP（简单双口）RAM。在这个模式下，DRM 端口数据位宽增大至 36 bits，A， B 两个端口中一个端口专用于数据写入，另一个端口专用于数据读取，读写端口同样均支持不同的时钟。与 DP RAM 相同的是，若同时通过两个端口对同一地址进行读写操作会引起冲突。读者通过回读表 3 和表 6 可分别查看 9Kb、18Kb 模式下 SDP RAM 允许与不允许的组合模式。如图 7 所示对 SDP RAM 数据流进行说明。表 11 罗列了端口名称和端口的描述。

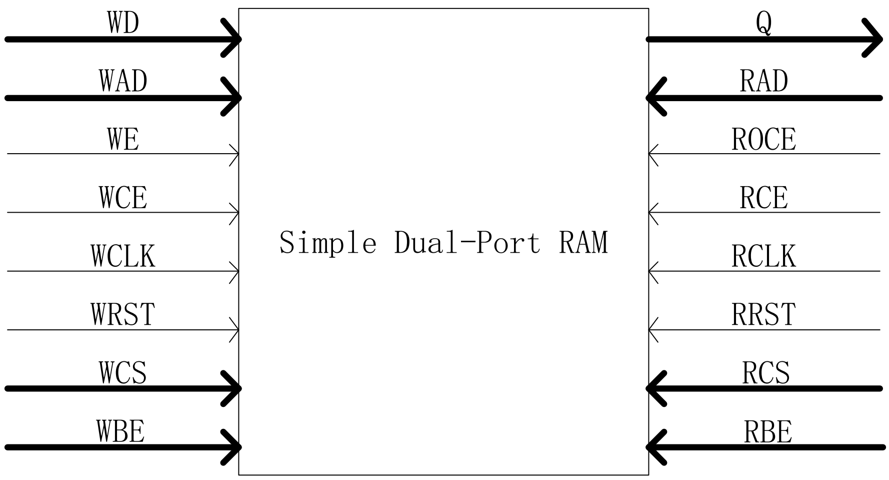


图 7 简单双端口 RAM 数据端口

表 11 SDP RAM 端口命名及描述

端口命名	输入/输出	描述	端口命名	输入/输出	描述
WD	输入	数据输入	Q	输出	数据输出
WAD	输入	写地址输入	RAD	输入	读地址输入
WE	输入	写使能	ROCE	输入	读出寄存器 OE
WCE	输入	写时钟使能	RCE	输入	读时钟使能
WCLK	输入	写时钟	RCLK	输入	读时钟
WRST	输入	写寄存器复位	RRST	输入	读寄存器复位
WCS	输入	写地址扩展	RCS	输入	读地址扩展
WBE	输入	写字节使能控制	RBE	输入	读字节使能控制

SDP RAM 的典型时序如下：

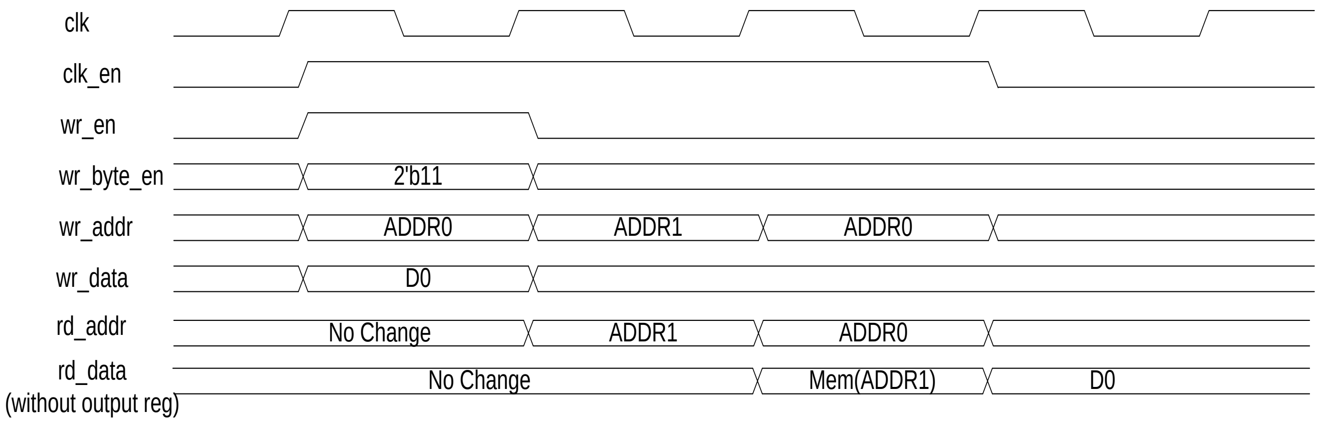


图 8 SDP RAM 典型时序图

四、SP RAM 模式和 ROM 模式

（一）SP RAM 模式数据端口

SP RAM 模式在 9Kb 模式下，DRM 包含两个端口，SP RAM 模式可以分别对这两个端口独立进行操作。

SP RAM 模式在两个端口共享情况下，DRM 仅包含一个端口，SP RAM 模式可以对这个端口进行读写操作。

如图 9 所示对 SP RAM 模式的数据流进行说明，表 12 罗列了端口名称和端口的描述。

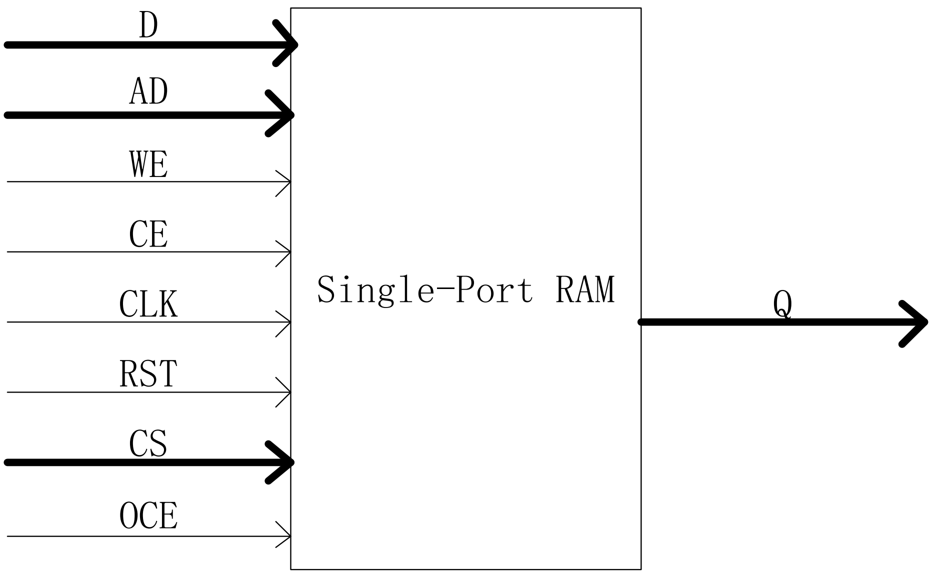


图 9 SP RAM 模式

表 12 SP RAM 模式端口命名及描述

端口命名	输入/输出	描述	端口命名	输入/输出	描述
D	输入	数据输入	RST	输入	写寄存器复位
AD	输入	写地址输入	CS	输入	写地址扩展
WE	输入	写使能	OCE	输入	读出寄存器 OE
CE	输入	写时钟使能	Q	输出	数据输出
CLK	输入	写时钟			

（二）ROM 模式数据接口

DRM 可以配置为 ROM，通过配置接口可以初始化 ROM 内容。由于没有内建逻辑去控制 ROM 模式，需要 PDS 软件禁止写操作。ROM 模式对 9Kb 模式或两端口共享模式的端口只读。

如图 10 所示对 ROM 模式的数据流进行说明，表 13 罗列了端口名称和端口的描述。

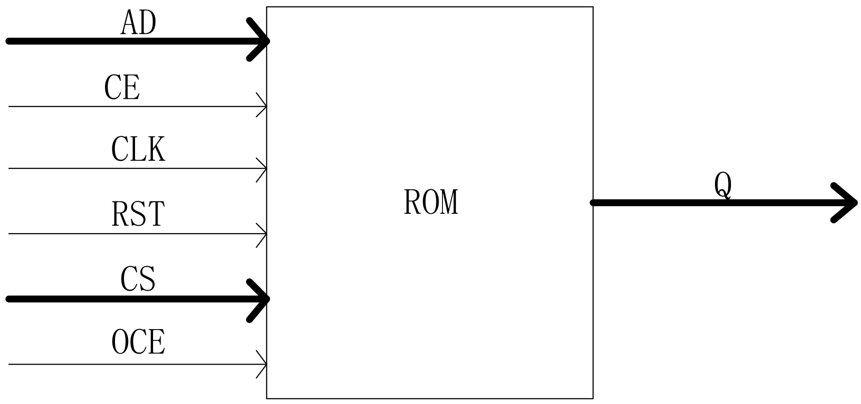


图 10 ROM 模式

表 13 ROM 模式端口命名及描述

端口命名	输入/输出	描述	端口命名	输入/输出	描述
AD	输入	写地址输入	CS	输入	写地址扩展
CE	输入	写时钟使能	OCE	输入	读出寄存器 OE
CLK	输入	写时钟	Q	输出	数据输出
RST	输入	写寄存器复位			

（三）读操作

SP RAM模式支持：

- 单端口读操作
- 单端口写操作
- x32/x36 位宽A端口固定为写入控制端口（WEA[0/1] 为高时（default极性）有效），B端口固定为读出控制端口（WEB[0/1] 为低时（default极性）有效）。

在 DRM 单元中，A/B 两个端口都包括有数据输出 Latch。在读操作中，输出寄存器（OR）旁路时，QA/QB 输出为 Latch 输出，在同一个读时钟周期时钟上沿（默认时钟极性）输出。

（四）写操作

SP RAM (x32/x36)模式下 TW 和 RBW 的实现

DRM 无法通过直接配置实现 32/36 bit SP 模式下的 TW（Transparent Write）和 RBW（Read before Write）模式，需要额外的配置，具体实现方式如下：

a. Port A 和 Port B 同时配置为 16/18 bit DP 模式（分别针对 32/36 bit SP）；

b. Port A 和 Port B 配置为相同的写模式；其他配置位也需要相同。

c. Port A 和 Port B 的 CE，CS，WE，ADS，OCE，RST，CLK 等控制信号分别并联起来；

d. DA、DB 并联作为数据输入；QA、QB 并联作为数据输出；

e. Port A 和 Port B 地址输入端口需同时接地址输入；ADA[4]和 ADB[4]需分别接 0 或者 1（相反），如图 11 所示：

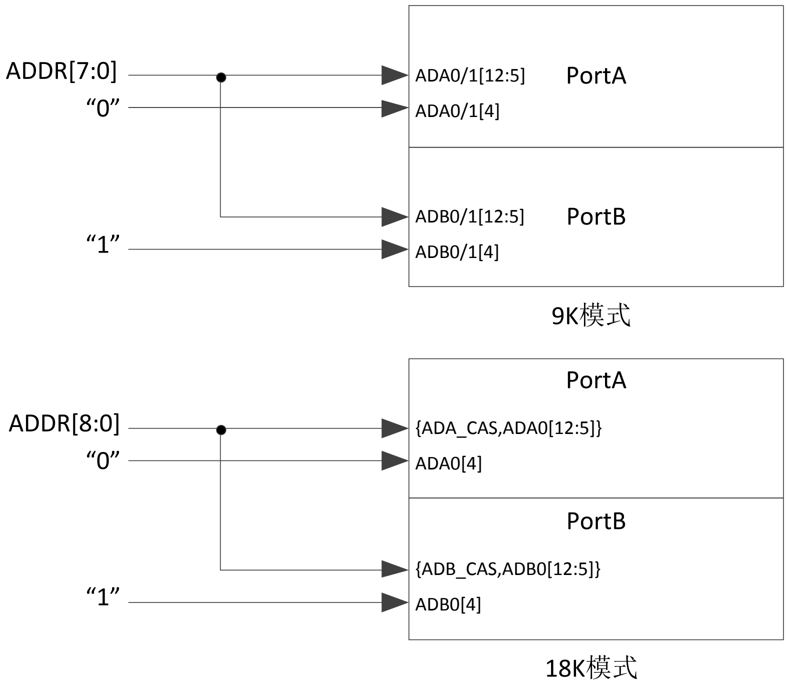


图 11 SP RAM 模式下 TW 和 RBW 模式地址线配置

f. Byte Enable[3:0]信号的连接需要分别参考两个端口 DP 16/18 bit 模式下的连接。

五、FIFO 模式

（一）FIFO 模式下端口映射

如图 12 所示为 FIFO 模式下各端口映射关系，此时所有控制端口极性为默认配置（注：默认为高电平有效，时钟为上升沿有效，详见（十）逻辑控制）。

此模式下，异步 FIFO 读写时钟域相互独立。写操作同步于写时钟而读操作同步于读时钟。

异步 FIFO 模式下，当写使能有效，且 FIFO 不为满时，在写时钟的上升沿向 FIFO 写入数据；当读使能有效，且 FIFO 不为空时，在读时钟的上升沿从 FIFO 读出数据。当 FIFO 写满时产生满信号，满信号同步于写时钟域。当 FIFO 读空时产生空信号，空信号同步于读时钟域。

同步 FIFO 模式下，读写时钟为同一个时钟。当写使能有效，且 FIFO 不为满时，在时钟的上升沿向 FIFO 写入数据；当读使能有效，且 FIFO 不为空时，在时钟的上升沿从 FIFO 读出数据。当 FIFO 写满时产生满信号，当 FIFO 读空时产生空信号。

DRM_FIFO 位宽模式详见附录。

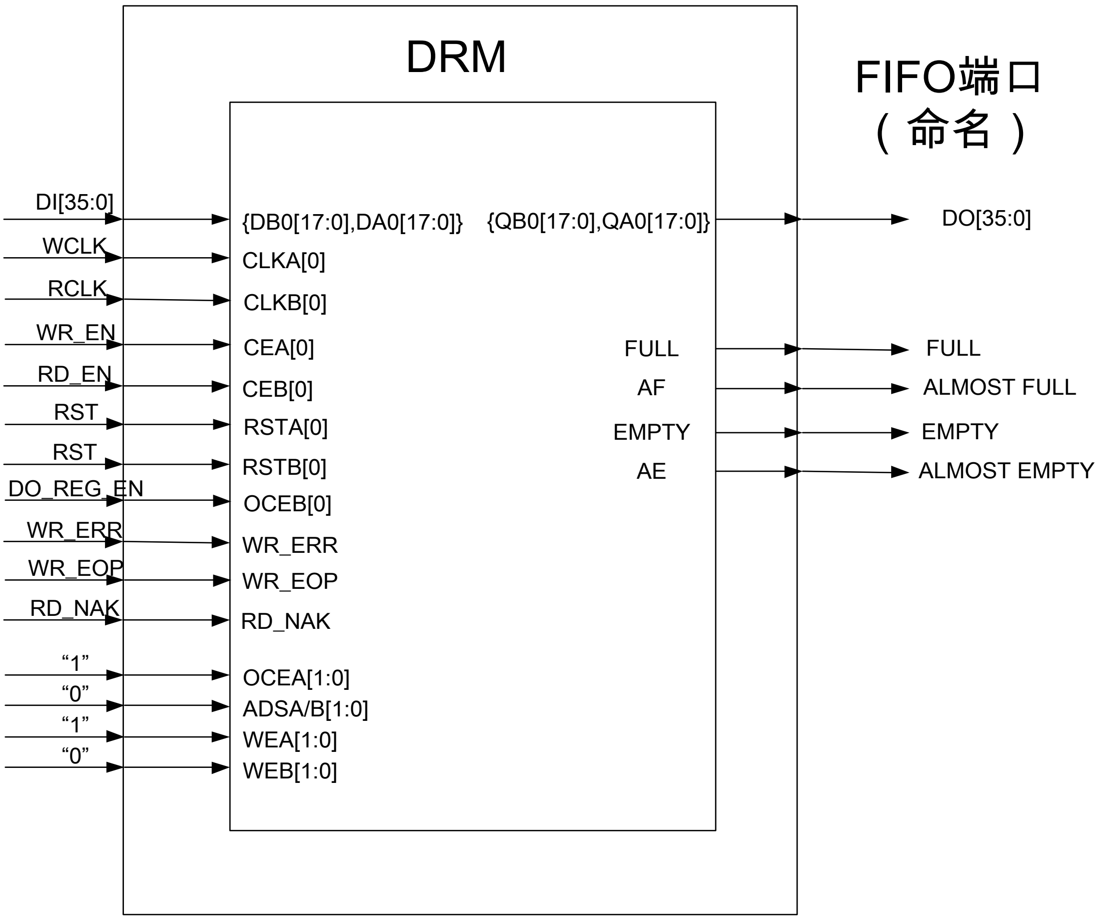


图 12 FIFO 模式下端口映射关系

（二）异步 FIFO（不含丢包重发功能）

此模式下数据流如图 13 所示：

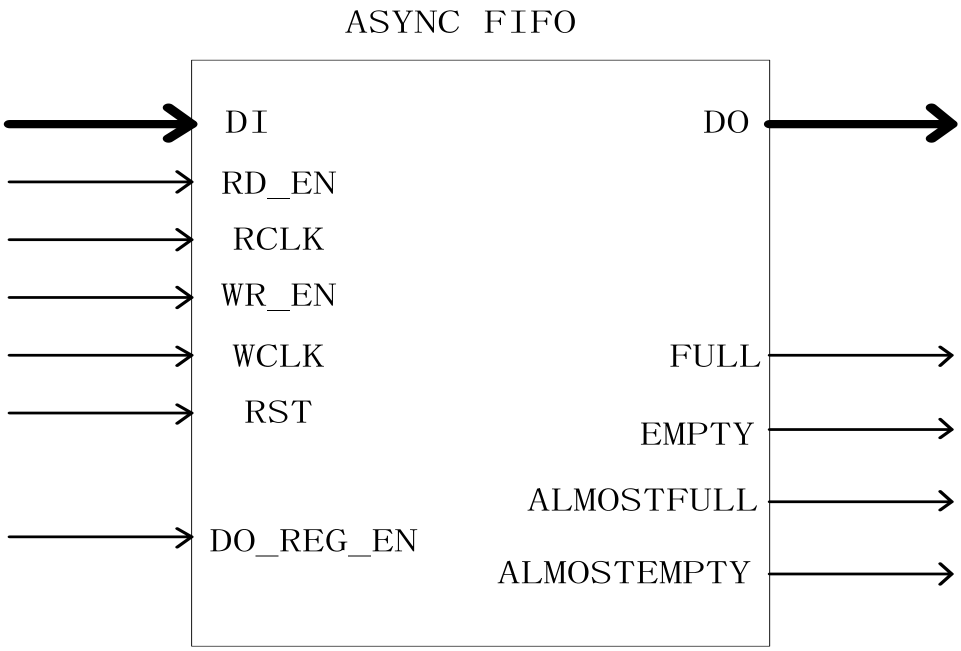


图 13 异步 FIFO 模式（不含丢包重发功能）

表 14 异步 FIFO 模式（不含丢包重发功能）端口说明

端口名	输入/输出	描述
DI	输入	数据输入
RD_EN	输入	读出数据使能信号，高有效
RCLK	输入	读出数据时钟信号
WR_EN	输入	写入数据使能信号，高有效
WCLK	输入	写入数据时钟信号
RST	输入	复位信号，高有效
DO_REG_EN	输入	读寄存器使能信号，高有效
DO	输出	数据输出
FULL	输出	满标志
EMPTY	输出	空标志
ALMOSTFULL	输出	将满标志
ALMOSTEMPTY	输出	将空标志

（三）异步 FIFO（支持丢包功能）

丢包功能：异步 FIFO 写入端检测到 wr_err 数据包错误信号时，写入端的写地址退回到数据包包头所在的写地址。丢包功能可以单独使能，wr_eop 为输入信号。

此模式下数据流如图 14 所示：

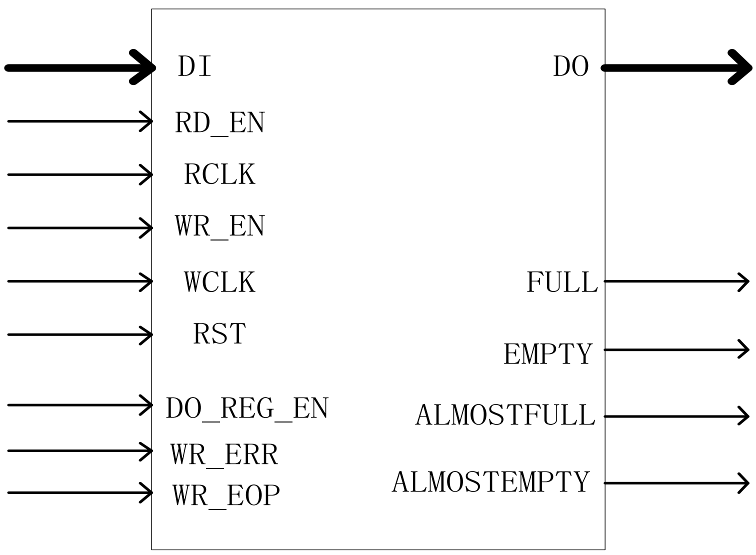


图 14 异步 FIFO 模式（支持丢包功能）

表 15 异步 FIFO 模式（支持丢包功能）端口说明

端口名	输入/输出	描述
DI[n:0]	输入	数据输入
RD_EN	输入	读出数据使能信号，高有效
RCLK	输入	读出数据时钟信号
WR_EN	输入	写入数据使能信号，高有效
WCLK	输入	写入数据时钟信号
RST	输入	复位信号，高有效
DO_REG_EN	输入	读寄存器使能信号，高有效
WR_ERR	输入	写端口数据包指示
WR_EOP	输入	包尾信号指示
DO[N:0]	输出	数据输出
FULL	输出	满标志
EMPTY	输出	空标志
ALMOSTFULL	输出	将满标志
ALMOSTEMPTY	输出	将空标志

（四）异步 FIFO（包含丢包重发功能）

丢包功能：异步 FIFO 写入端检测到 wr_err 数据包错误信号时，写入端的写地址退回到数据包包头所在的写地址。

重发功能：当异步 FIFO 读时钟检测到 rd_nak 信号，读地址退回到数据包包头所在的地址。

丢包功能和重发功能需要同时使能。wr_eop 输入连接到数据输入 DI。

此模式下数据流如图 15 所示：

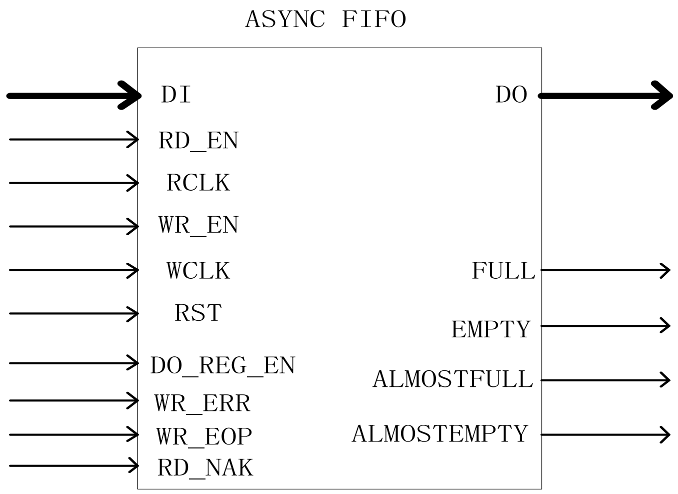


图 15 异步 FIFO 模式（包含丢包重发功能）

表 16 异步 FIFO 模式（包含丢包重发功能）端口说明

端口名	输入/输出	描述
DI	输入	数据输入
RD_EN	输入	读出数据使能信号，高有效
RCLK	输入	读出数据时钟信号
WR_EN	输入	写入数据使能信号，高有效
WCLK	输入	写入数据时钟信号
RST	输入	复位信号，高有效
DO_REG_EN	输入	读寄存器使能信号，高有效
WR_ERR	输入	写端口数据包指示
WR_EOP	输入	包尾信号指示
RD_NAK	输入	读端口重发指示
DO	输出	数据输出
FULL	输出	满标志
EMPTY	输出	空标志
ALMOSTFULL	输出	将满标志
ALMOSTEMPTY	输出	将空标志

（五）同步 FIFO（不含丢包重发功能）

同步 FIFO 读写时钟为同一个时钟。

当写使能有效，且 FIFO 不为满时，在时钟的上升沿向 FIFO 写入数据。

当读使能有效，且 FIFO 不为空时，在时钟的上升沿从 FIFO 读出数据。

当 FIFO 写满时产生满信号，当 FIFO 读空时产生空信号。

此模式下数据流如图 16 所示：

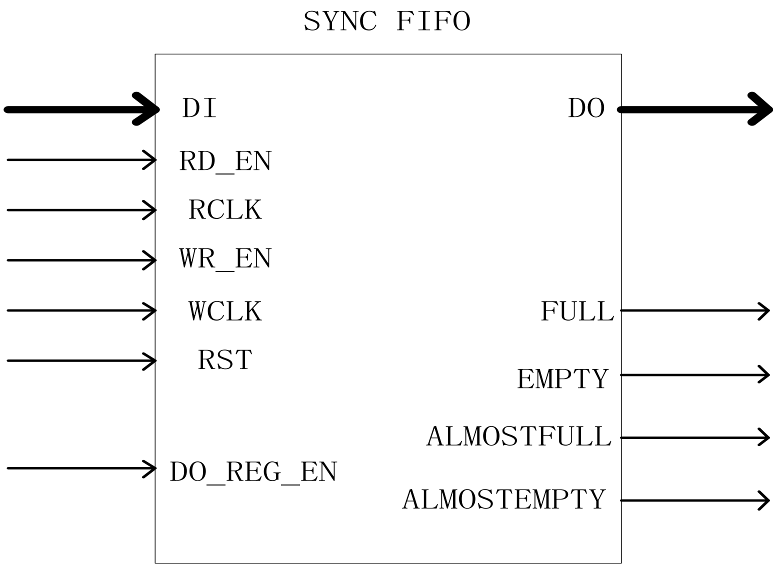


图 16 同步 FIFO 模式（不含丢包重发功能）

表 17 同步 FIFO 模式（不含丢包重发功能）端口说明

端口名	输入/输出	描述
DI	输入	数据输入
RD_EN	输入	读出数据使能信号，高有效
RCLK	输入	读出数据时钟信号
WR_EN	输入	写入数据使能信号，高有效
WCLK	输入	写入数据时钟信号
RST	输入	复位信号，高有效
DO_REG_EN	输入	读寄存器使能信号，高有效
DO	输出	数据输出
FULL	输出	满标志
EMPTY	输出	空标志
ALMOSTFULL	输出	将满标志
ALMOSTEMPTY	输出	将空标志

（六）同步 FIFO（支持丢包重发功能）

丢包功能：同步 FIFO 写入端检测到 wr_err 数据包错误信号时，写入端的写地址退回到数据包包头所在的写地址。丢包功能可以单独使能，wr_eop 为输入信号。

此模式下数据流如图 17 所示：

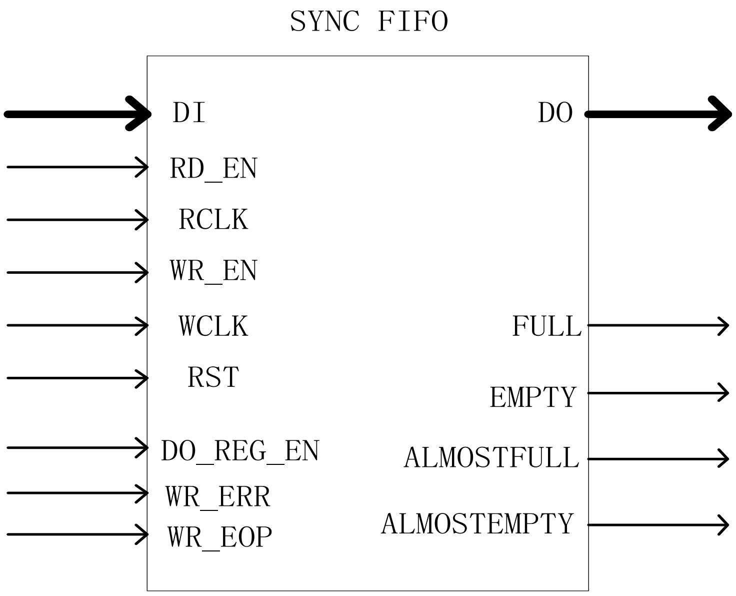


图 17 同步 FIFO 模式（支持丢包功能）

表 18 同步 FIFO 模式（支持丢包功能） 端口说明

端口名	输入/输出	描述
DI	输入	数据输入
RD_EN	输入	读出数据使能信号，高有效
RCLK	输入	读出数据时钟信号
WR_EN	输入	写入数据使能信号，高有效
WCLK	输入	写入数据时钟信号
RST	输入	复位信号，高有效
DO_REG_EN	输入	读寄存器使能信号，高有效
WR_ERR	输入	写端口数据包指示
WR_EOP	输入	包尾信号指示
DO	输出	数据输出
FULL	输出	满标志
EMPTY	输出	空标志
ALMOSTFULL	输出	将满标志
ALMOSTEMPTY	输出	将空标志

（七）同步 FIFO（包含丢包重发功能）

丢包功能： FIFO 写入端检测到 wr_err 数据包错误信号时，写入端的写地址退回到数据包包头所在的写地址。

重发功能：当同步 FIFO 读时钟检测到 rd_nak 信号，读地址退回到数据包包头所在的地址。

丢包功能和重发功能需要同时使能。wr_eop 输入连接到数据输入 DI。

此模式下数据流如图 18 所示：

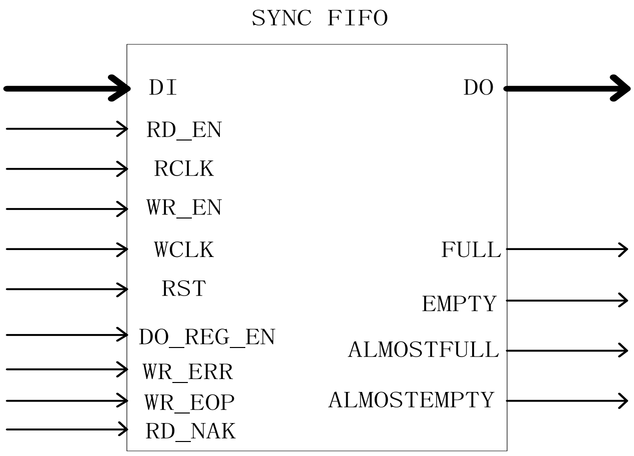


图 18 同步 FIFO（包含丢包重发功能）

表 19 同步 FIFO（包含丢包重发功能）端口说明

端口名	输入/输出	描述
DI[n:0]	输入	数据输入
RD_EN	输入	读出数据使能信号，高有效
RCLK	输入	读出数据时钟信号
WR_EN	输入	写入数据使能信号，高有效
WCLK	输入	写入数据时钟信号
RST	输入	复位信号，高有效
DO_REG_EN	输入	读寄存器使能信号，高有效
WR_ERR	输入	写端口数据包指示
WR_EOP	输入	包尾信号指示
RD_NAK	输入	读端口重发指示
DO[N:0]	输出	数据输出
FULL	输出	满标志
EMPTY	输出	空标志
ALMOSTFULL	输出	将满标志
ALMOSTEMPTY	输出	将空标志

六、DRM_FIFO 读写时序

（一）写入空 FIFO 时序

empty 信号指示 FIFO 为空，当 wr_en 有效并且有效数据写入后，同步 FIFO 时，empty 信号在 1 个 rclk 时钟周期后置“0”；写入异步 FIFO 时，empty 信号在 2 个 rclk 时钟周期后置零。当 wr_en 继续有效，almost_empty 信号根据 almost_empty_offset 的配置延迟置零。

写入空 FIFO 时序图如图 19 所示：

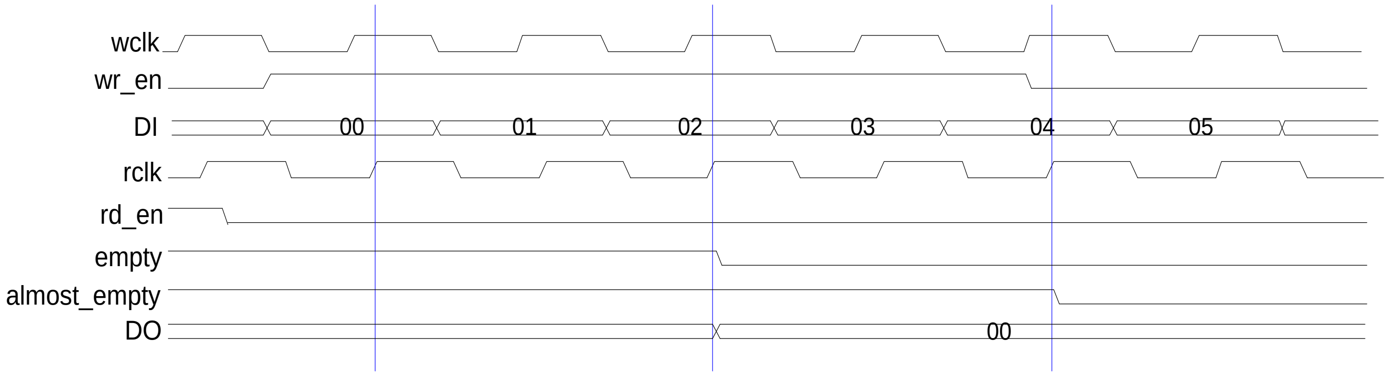


图 19 写入空 FIFO 时序图

（二）写入将满 FIFO 时序

FIFO 将满时，根据 almost_full_offset 的配置，almost_full 将提前置 1。当 FIFO full 时，写指针不再增加。

写入将满 FIFO 时序图如

图 20 所示：

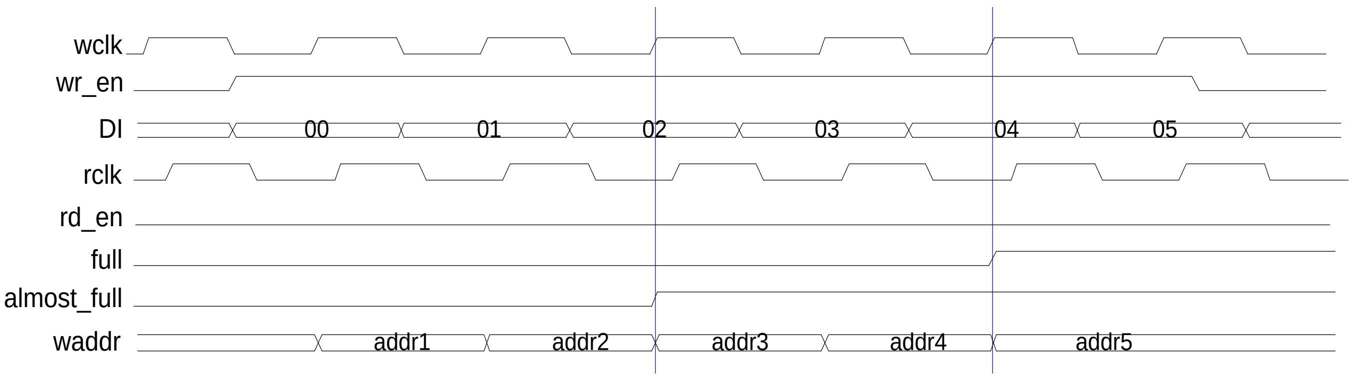


图 20 写入将满 FIFO 时序图

（三）满 FIFO 读出时序

full 信号指示 FIFO 为满，当 rd_en 有效读出数据后，full 信号在 1 到 2 个 wclk 时钟周期内置零。当 rd_en 继续有效，almost_full 信号根据 almost_full_offset 的配置延迟置零。从满 FIFO 读出时序图如图 21 所示：

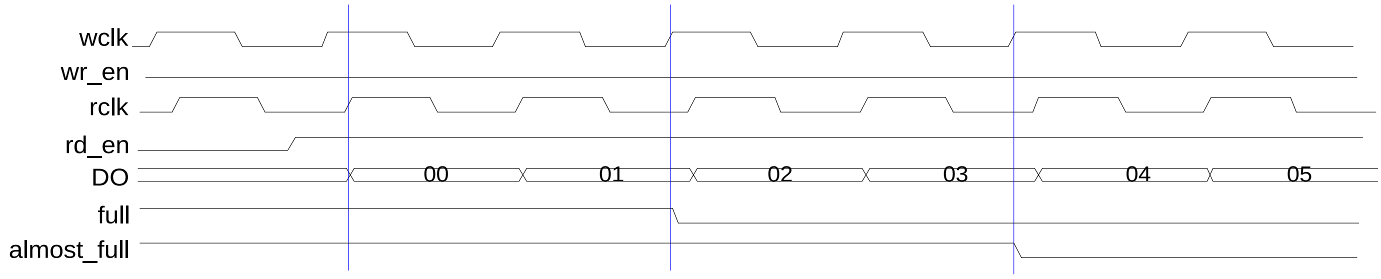


图 21 从满 FIFO 读出时序图

（四）将空 FIFO 读出时序

FIFO 将空时，根据 almost_empty_offset 的配置，almost_empty 提前置 1。当 FIFO empty 时，读指针不再增加。从将空 FIFO 读出时序图如

图 22 所示：

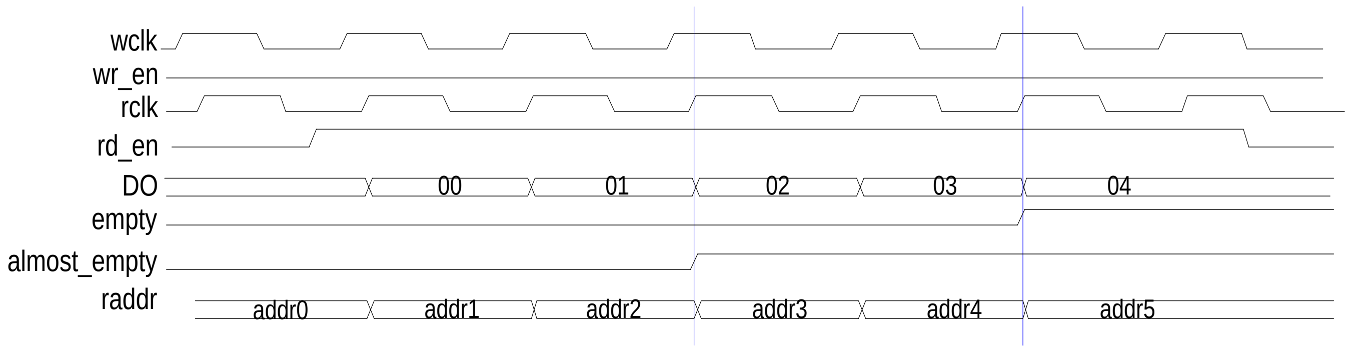


图 22 从将空 FIFO 读出时序

（五）写丢包时序

当包尾指示信号 wr_eop 置“1”并且 wr_err 数据包错误指示有效时，写指针在下一个 wclk 周期退回数据包包头所在位置。写丢包时序图如

图 23 所示：

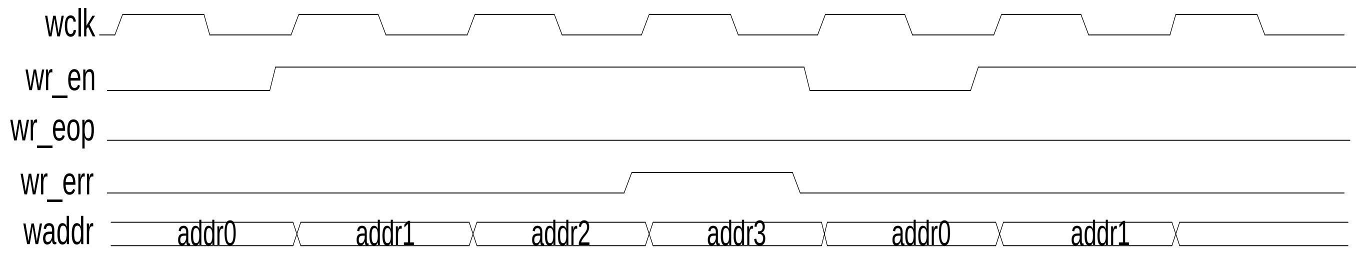


图 23 写丢包时序

（六）读重发时序

当包尾指示信号 wr_eop 置“1”时标记上一组数据包包尾地址，读指针在 rd_nak 信号置“1”后退回到当前数据包包头所在地址。

写重发时序如图 24 所示：

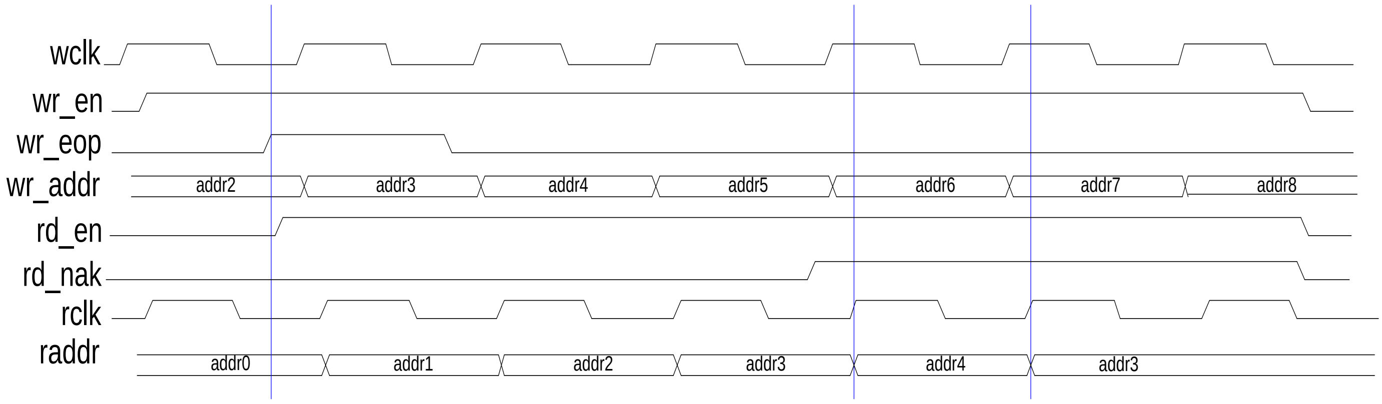


图 24 读重发时序

（七）支持丢包功能下写入空 FIFO

empty 信号指示 FIFO 为空，当 wr_en 有效，并且第一个 wr_eop 包尾信号有效（wr_err 等于 0），empty 信号在 rclk 时钟域内置“0”。

写丢包功能下写入空 FIFO 时序图如图 25 所示：

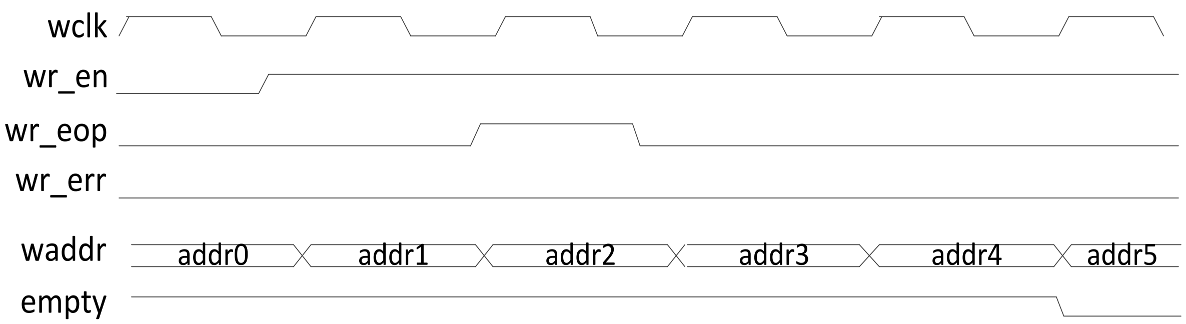


图 25 写丢包模式下写入空 FIFO 时序图

（八）丢包重发功能下写入将满 FIFO

full 信号指示 FIFO 写满，当读端口读出 wr_eop 所在读地址的数据且 rd_nak 为“0”时，表示读出包数据有效。当读出数据未满一个包，并且写指针指向最后一个读出有效包尾地址时，会产生 full 信号。

丢包重发模式下写入将满 FIFO 时序图如图 26 所示：



图 26 丢包重发模式下写入将满 FIFO 时序图

七、寄存器模式

DRM 支持输入寄存器（IR）、输出寄存器（OR）以及 Core Latch。其结构如图 27 所示：

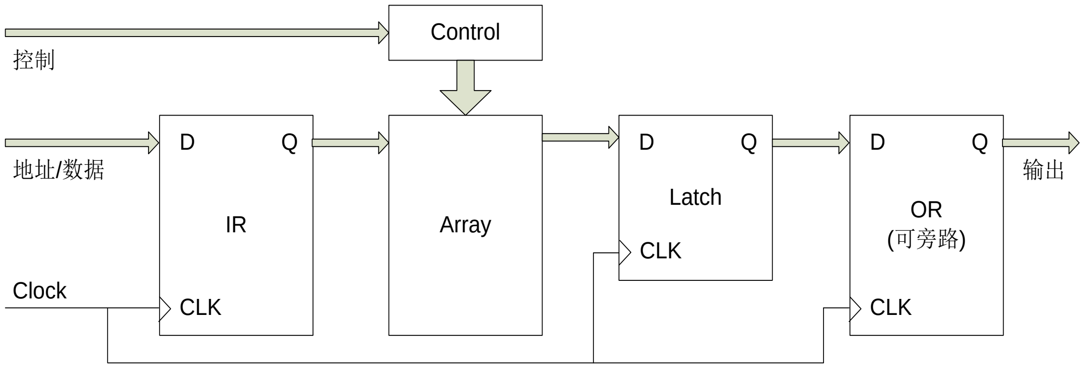


图 27 DRM 寄存器

（一）输入寄存器（IR）

端口地址、数据、部分控制信号都有相应的输入寄存器（IR），通过 IR 可以实现同步写操作。Logos 系列 FPGA 的 DRM 的 IR 总数为 134，表 20 为 IR 的端口输入列表：

表 20 端口输入 IR 列表

端口输入	寄存器	端口输入	寄存器
ADA0[12:0]	IR	WEB[1:0]	IR
DA0[17:0]	IR	ADA1[12:0]	IR
CSA[2:0]	IR	DA1[17:0]	IR
WEA[1:0]	IR	ADB1[12:0]	IR
ADB0[12:0]	IR	DB1[17:0]	IR
DB0[17:0]	IR	ADA_CAS	IR
CSB[2:0]	IR	ADB_CAS	IR

在地址输入端口 ADA0[12:0]、ADA1[12:0]、ADB0[12:0]、ADB1[12:0]、ADA_CAS、ADB_CAS，内置地址锁存选择通路。其逻辑实现如图 28 所示，时序图如图 29 所示。

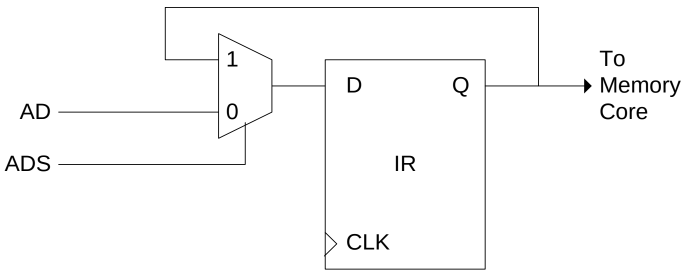


图 28 地址 IR 逻辑

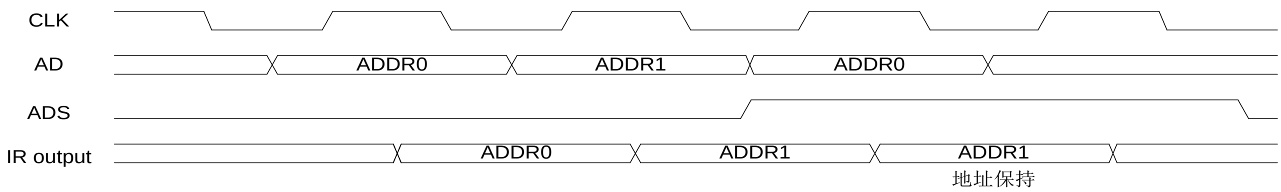


图 29 地址 IR 时序

（二）输出寄存器（OR）

每个 9K 模块在输出端口 QA[17:0]和 QB[17:0]带有可旁路的输出寄存器（OR），Logos 系列 FPGA 的 DRM 的 OR 总数为 36。通过 OR 可以实现 Pipeline 设计以相应提高设计性能。如图 30 所示为 OR 逻辑图：

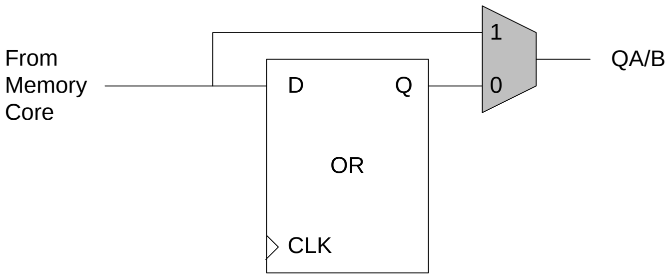


图 30 OR 逻辑图

（三）Core Latch

在 Logos 系列 FPGA 的 DRM 单元中，A/B 两个端口都包括有数据输出 Latch。在读操作中，输出寄存器（OR）旁路时，QA/QB 输出为 Latch 输出，在同一个读时钟周期时钟上沿（默认时钟极性）输出。

（四）Pipeline 模式

Pipeline 模式寄存器读/写操作。在该模式下，在 A/B 端口对应的配置位可以为旁路输出寄存器。所以根据工作模式有两种 Pipeline 情况：

① 旁路输出寄存器同步写，在存储器 Core 可以锁存输出，如图 31 所示。

② 使能输出寄存器同步写，输出寄存器支持最大位宽为 36 位的同步读出，如图 32 所示。

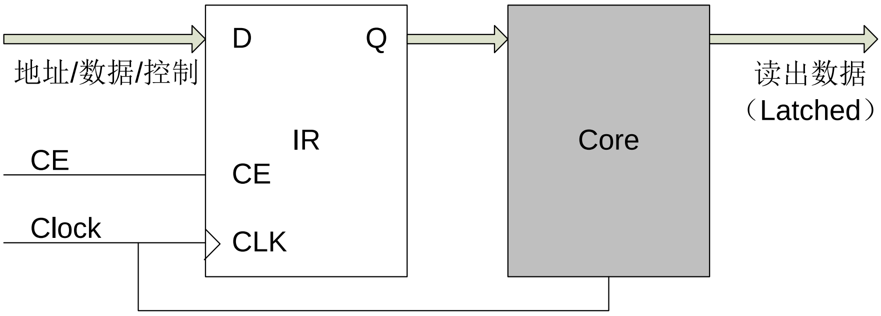


图 31 旁路输出寄存器 Pipeline 模式

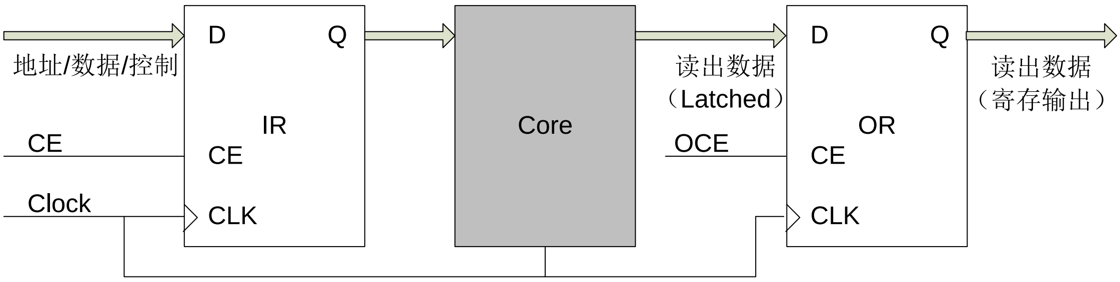


图 32 使能输出寄存器 Pipeline 模

八、混合数据位宽功能

DRM 的端口位宽支持两种类型：一种是数据位宽为2^N（包括 1/2/4/8/16/32bit）。另一种是数据位宽为9×2^N（包括 9/18/36bit）。

DP RAM 和 SDP RAM 模式支持混合数据位宽功能，即两个端口可以配置成不同位宽。例如，一个 SDP RAM，可以在写端口配置成 16Kx1，在读端口配置成 512x32，从而节约了从 1 比特到 32 比特的串并转换逻辑。

九、DRM 级联扩展功能

多个 DRM 可以通过级联扩展的方式组合成更大的 DP RAM，SDP RAM, SP RAM，ROM 或者 FIFO。对此，DRM 提供额外的 3 bit 地址扩展（CS[2:0]），常用于深度扩展的应用。

十、控制逻辑

（一）输入极性控制

Port A/Port B 两个端口的控制信号在 DRM 本地都有极性控制。如表 21 所示：

表 21 输入极性控制列表

输入信号	极性控制	Default 极性	输入信号	极性控制	Default 极性
CEA[1:0]	DRM 本地	高有效	CEB[1:0]	DRM 本地	高有效
CLKA[1:0]	DRM 本地	上升沿有效	CLKB[1:0]	DRM 本地	上升沿有效
CSA[2:0]	DRM 本地	高有效	CSB[2:0]	DRM 本地	高有效
WEA[1:0]	DRM 本地	高有效	WEB[1:0]	DRM 本地	高有效
OCEA[1:0]	DRM 本地	高有效	OCEB[1:0]	DRM 本地	高有效
RSTA[1:0]	DRM 本地	高有效	RSTB[1:0]	DRM 本地	高有效

注：表格中各输入信号都有各自独立的极性控制

（二）同步、异步复位

如图 33 所示，DRM 输出（包括 Core 中 Latch 输出和 OR 输出）支持同步或者异步复位：

同步复位：RSTA/B[0/1]同步于 CLKA/B[10]，需满足对时钟沿的建立/保持时间；

异步复位：RSTA/B[0/1]独立于 CLKA/B[1/0]，输出复位于 RSTA/B 的边沿。

A/B 两个端口共用同步/异步复位选择设置。

异步复位同步复位失效，当 RSTA/B 有效时，独立于时钟复位内部 Latch 以及输出寄存器，当 RSTA/B 失效时，内部 Latch 以及输出寄存器同步于时钟沿恢复正常功能。

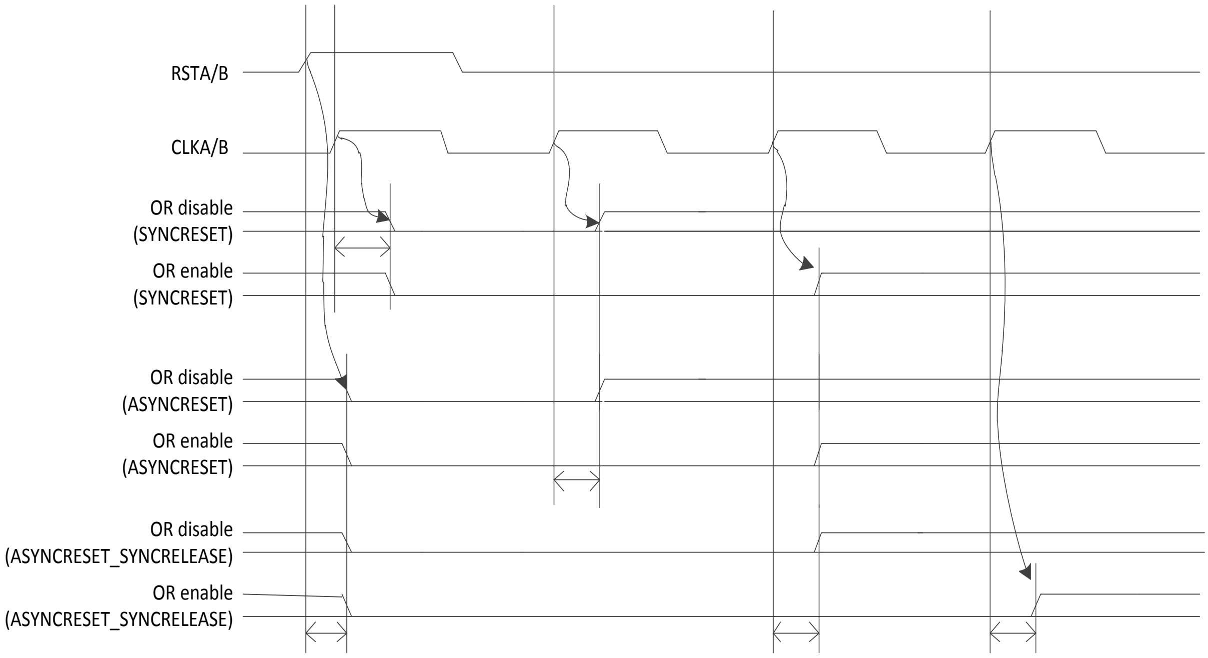


图 33 输出寄存器复位时序

（三） Core Latch 复位

如图 34 所示在 Logos 系列 FPGA 的 DRM 单元中，A/B 两个端口都包括有数据输出 Latch。这些 Latch 可以被同步或异步复位。其复位控制来自全局复位 GRSN 或者本地端口复位 RSTA/B。

注：GSR_N 为异步复位。GSR_N 复位可以通过配置位禁止。

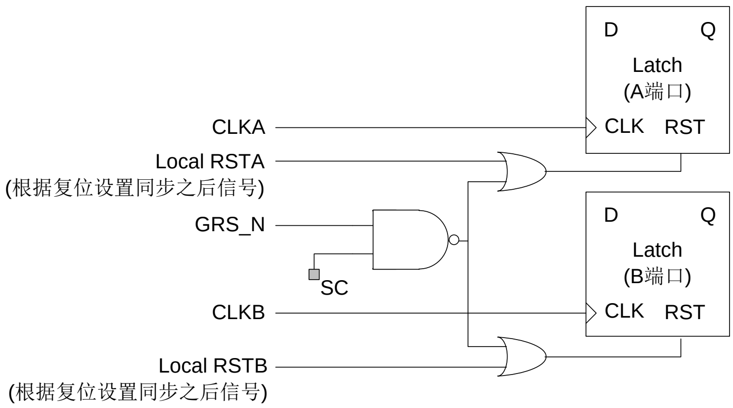


图 34 Core 复位

十一、全局信号时序

如图 35 所示，T0 至 T1 之间：下载位流，包括 sc_bit 和 rom 初始化内容；T1 时刻之前，DRM 所有输出为 1；GLOGEN 释放后，DRM 所有输出状态，与配置和输入有关；

GRS_N 和 GWEN 以及 DONE（DRM 中不可用）的释放时刻，用户通过 CCS 可分别独立配置，可配置在 T2 或 T3 或 T4 时刻；在 DONE 信号置 1 后，进入用户模式。

全局信号详细说明请参考 PGL CCS TSPEC 文档

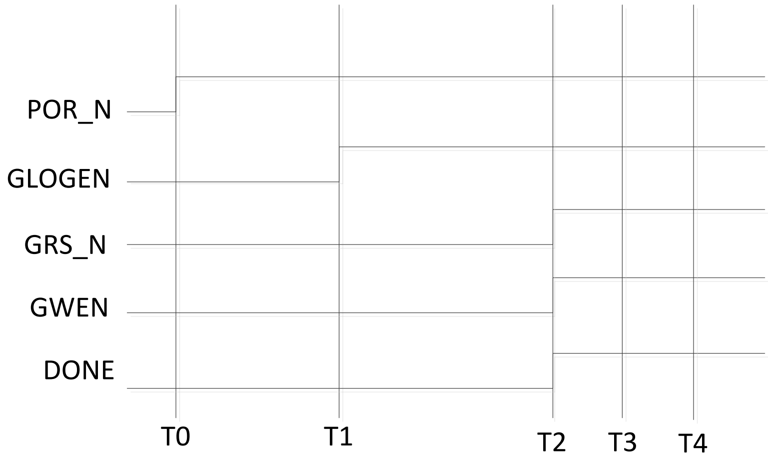


图 35 全局信号时序示意图

附录

地址和数据端口 Mapping

如表 22、表 23、表 24 所示分别为 18Kb 和 9Kb 模块 0、9Kb 模块 1 存储器模式下，对应的地址和数据端口 Mapping。

注:1. Logos系列DRM具有单个18Kbits或者两个9Kbits存储空间，故9Kb有模块0和模块1对应的地址和数据端口Mapping。

表 22 18Kb DRM 模式下存储器对应的地址和数据端口 Mapping

组合模式	A 端口地址	B 端口地址	A 端口数据输入	B 端口数据输入	A 端口数据输出	B 端口数据输出
18Kb 的 DRM 存储器模式						
16K*1	{ADA_CAS, ADA0[12:0]}	{ADB_CAS, ADB0[12:0]}	DA0[0]	DB0[0]	QA0[X],X=16~9, 7~0 中任意一位	QB0[X],X=16~9, 7~0 中任意一位
8K*2	{ADA_CAS, ADA0[12:1]}	{ADB_CAS, ADB0[12:1]}	DA0[1:0]	DB0[1:0]	QA0[X+1:X],X= 0,2,4,9,11,12,15	QB0[X+1:X],X= 0,2,4,9,11,12,15
4K*4	{ADA_CAS, ADA0[12:2]}	{ADB_CAS, ADB0[12:2]}	DA0[3:0]	DB0[3:0]	QA0[X+3:X],X= 0,4,9,12	QB0[X+3:X],X= 0,4,9,12
2K*8	{ADA_CAS, ADA0[12:3]}	{ADB_CAS, ADB0[12:3]}	DA0[7:0]	DB0[7:0]	QA0[7:0] 或者 QA0[16:9]	QB0[7:0] 或者 QB0[16:9]
1K*16	{ADA_CAS, ADA0[12:4]}	{ADB_CAS, ADB0[12:4]}	DA0[16:9,7:0]	DB0[16:9,7:0]	QA0[16:9,7:0]	QB0[16:9,7:0]
2K*9	{ADA_CAS, ADA0[12:3]}	{ADB_CAS, ADB0[12:3]}	DA0[8:0]	DB0[8:0]	QA0[8:0] 或者 QA0[17:9]	QB0[8:0] 或者 QB0[17:9]
1K*18	{ADA_CAS, ADA0[12:4]}	{ADB_CAS, ADB0[12:4]}	DA0[17:0]	DB0[17:0]	QA0[17:0]	QB0[17:0]
512*32	{ADA_CAS, ADA0[12:5]}	{ADB_CAS, ADB0[12:5]}	{DB0[16:9,7:0], DA0[16:9,7:0]}	N/A	N/A	{QB0[16:9,7:0], QA0[16:9,7:0]}
512*36	{ADA_CAS, ADA0[12:5]}	{ADB_CAS, ADB0[12:5]}	{DB0[17:0], DA0[17:0]}	N/A	N/A	{QB0[17:0], QA0[17:0]}

表 23 9Kb DRM 模式下模块 0 的存储器对应的地址和数据端口 Mapping

组合模式	A 端口地址	B 端口地址	A 端口数据输入	B 端口数据输入	A 端口数据输出	B 端口数据输出
DRM 端口模式(9K 模块 0)						
8K*1	ADA0[12:0]	ADB0[12:0]	DA0[0]	DB0[0]	QA0[X],X=16~9, 7~0 中任意一位	QB0[X],X=16~9, 7~0 中任意一位
4K*2	ADA0[12:1]	ADB0[12:1]	DA0[1:0]	DB0[1:0]	QA0[X+1:X],X= 0,2,4,9,11,12,15	QB0[X+1:X],X= 0,2,4,9,11,12,15
2K*4	ADA0[12:2]	ADB0[12:2]	DA0[3:0]	DB0[3:0]	QA0[X+3:X],X= 0,4,9,12	QB0[X+3:X],X= 0,4,9,12
1K*8	ADA0[12:3]	ADB0[12:3]	DA0[7:0]	DB0[7:0]	QA0[7:0] 或 者 QA0[16:9]	QB0[7:0] 或 者 QB0[16:9]
1K*9	ADA0[12:3]	ADB0[12:3]	DA0[8:0]	DB0[8:0]	QA0[8:0] 或 者 QA0[17:9]	QB0[8:0] 或 者 QB0[17:9]
512*16	ADA0[12:4]	ADB0[12:4]	DA0[16:9,7:0]	DB0[16:9,7:0]	QA0[16:9,7:0]	QB0[16:9,7:0]
512*18	ADA0[12:4]	ADB0[12:4]	DA0[17:0]	DB0[17:0]	QA0[17:0]	QB0[17:0]
256*32	ADA0[12:5]	ADB0[12:5]	DB0[16:9,7:0], DA0[16:9,7:0]}	N/A	N/A	QB0[16:9,7:0], QA0[16:9,7:0]}
256*36	ADA0[12:5]	ADB0[12:5]	{DB0[17:0], DA0[17:0]}	N/A	N/A	{QB0[17:0], QA0[17:0]}

表 24 9Kb DRM 模式下模块 1 的存储器对应的地址和数据端口 Mapping

组合模式	A 端口地址	B 端口地址	A 端口数据输入	B 端口数据输入	A 端口数据输出	B 端口数据输出
DRM 端口模式(9K 模块 1)						
8K*1	ADA1[12:0]	ADB1[12:0]	DA1[0]	DB1[0]	QA1[X],X=16~9, 7~0 中任意一位	QB1[X],X=16~9, 7~0 中任意一位
4K*2	ADA1[12:1]	ADB1[12:1]	DA1[1:0]	DB1[1:0]	QA1[X+1:X],X= 0,2,4,9,11,12,15	QB1[X+1:X],X= 0,2,4,9,11,12,15
2K*4	ADA1[12:2]	ADB1[12:2]	DA1[3:0]	DB1[3:0]	QA1[X+3:X],X= 0,4,9,12	QB1[X+3:X],X= 0,4,9,12
1K*8	ADA1[12:3]	ADB1[12:3]	DA1[7:0]	DB1[7:0]	QA1[7:0] 或者 QA1[16:9]	QB1[7:0] 或者 QB1[16:9]
1K*9	ADA1[12:3]	ADB1[12:3]	DA1[8:0]	DB1[8:0]	QA1[8:0] 或者 QA1[17:9]	QB1[8:0] 或者 QB1[17:9]
512*16	ADA1[12:4]	ADB1[12:4]	DA1[16:9,7:0]	DB1[16:9, 7:0]	QA1[16:9,7:0]	QB1[16:9,7:0]
512*18	ADA1[12:4]	ADB1[12:4]	DA1[17:0]	DB1[17:0]	QA1[17:0]	QB1[17:0]
256*32	ADA1[12:5]	ADB1[12:5]	DB1[16:9,7:0], DA1[16:9,7:0]}	N/A	N/A	QB1[16:9,7:0], QA1[16:9,7:0]}
256*36	ADA1[12:5]	ADB1[12:5]	{DB1[17:0], DA1[17:0]}	N/A	N/A	{QB1[17:0], QA1[17:0]}

如表 25、表 26 所示分别为 x1，x2，x4，x8，x16，x32 数据位宽和 x9，x18，x36 数据位宽的数据地址 Mapping。

表 25 不同位宽数据地址 Mapping（x1，x2，x4，x8，x16，x32 数据宽度）

<table>
<tr><th>数 据 位宽</th><th colspan="32">最低位端口地址 (与最大数据宽度模式最低位端口地址比较)</th></tr>
<tr><td>32</td><td colspan="32">0</td></tr>
<tr><td>16</td><td colspan="16">1</td><td colspan="16">0</td></tr>
<tr><td>8</td><td colspan="8">3</td><td colspan="8">2</td><td colspan="8">1</td><td colspan="8">0</td></tr>
<tr><td>4</td><td colspan="4">7</td><td colspan="4">6</td><td colspan="4">5</td><td colspan="4">4</td><td colspan="4">3</td><td colspan="4">2</td><td colspan="4">1</td><td colspan="4">0</td></tr>
<tr><td>2</td><td colspan="2">15</td><td colspan="2">14</td><td colspan="2">13</td><td colspan="2">12</td><td colspan="2">11</td><td colspan="2">10</td><td colspan="2">9</td><td colspan="2">8</td><td colspan="2">7</td><td colspan="2">6</td><td colspan="2">5</td><td colspan="2">4</td><td colspan="2">3</td><td colspan="2">2</td><td colspan="2">1</td><td colspan="2">0</td></tr>
<tr><td>1</td><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
</table>

表 26 不同位宽数据地址 Mapping（x9，x18，x36 数据宽度）

<table>
<tr><th>数据位宽</th><th colspan="36">最低位端口地址 (与最大数据宽度模式最低位端口地址比较)</th></tr>
<tr><td>36</td><td colspan="36">0</td></tr>
<tr><td>18</td><td colspan="18">1</td><td colspan="18">0</td></tr>
<tr><td>9</td><td colspan="9">3</td><td colspan="9">2</td><td colspan="9">1</td><td colspan="9">0</td></tr>
<tr><td>Index</td><td>35</td><td>34</td><td>33</td><td>32</td><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
</table>

字节附加信息位

在数据宽度为x9、x18、x36模式下，对应每字节有一位附加信息存储位，如表27所示：

表 27 字节附加信息位列表

	A 端口数据输入		**B 端口数据输入**		**A 端口数据输出**		**B 端口数据输出**	
端口模式	字节	附加信息位	字节	附加信息位	字节	附加信息位	字节	附加信息位
2K*9	DA0[7:0]	DA0[8]	DB0[7:0]	DB0[8]	QA0[16:9] 或 QA0[7:0]	QA0[17] 或 QA0[8]	QB0[16:9] 或 QB0[7:0]	QB0[17] 或 QB0[8]
1K*18	DA0[16:9]	DA0[17]	DB0[16:9]	DB0[17]	QA[16:9]	QA0[17]	QB0[16:9]	QB0[17]
	DA0[7:0]	DA0[8]	DB0[7:0]	DB0[8]	QA0[7:0]	QA0[8]	QB0[7:0]	QB0[8]
512*36	DA0[16:9]	DA0[17]	N/A	N/A	N/A	N/A	QA0[16:9]	QA0[17]
	DA0[7:0]	DA0[8]	N/A	N/A	N/A	N/A	QA0[7:0]	QA0[8]
	DB0[16:9]	DB0[17]	N/A	N/A	N/A	N/A	QB0[16:9]	QB0[17]
	DB0[7:0]	DB0[8]	N/A	N/A	N/A	N/A	QB0[7:0]	QB0[8]
1K*9(9K 模块 0)	DA0[7:0]	DA0[8]	DB0[7:0]	DB0[8]	QA0[16:9] 或 QA0[7:0]	QA0[17] 或 QA0[8]	QB0[16:9] 或 QB0[7:0]	QB0[17] 或 QB0[8]
1K*9(9K 模块 1)	DA1[7:0]	DA1[8]	DB1[7:0]	DB1[8]	QA1[16:9] 或 QA1[7:0]	QA1[17] 或 QA1[8]	QB1[16:9] 或 QB1[7:0]	QB1[17] 或 QB1[8]
512*18(9K 模块 0)	DA0[16:9]	DA0[17]	DB0[16:9]	DB0[17]	QA[16:9]	QA0[17]	QB0[16:9]	QB0[17]
	DA0[7:0]	DA0[8]	DB0[7:0]	DB0[8]	QA0[7:0]	QA0[8]	QB0[7:0]	QB0[8]
512*18(9K 模块 1)	DA1[16:9]	DA1[17]	DB1[16:9]	DB1[17]	QA[16:9]	QA1[17]	QB1[16:9]	QB1[17]
	DA1[7:0]	DA1[8]	DB1[7:0]	DB1[8]	QA1[7:0]	QA1[8]	QB1[7:0]	QB1[8]
256*36(9K 模块 0)	DA0[16:9]	DA0[17]	N/A	N/A	N/A	N/A	QB0[16:9]	QB0[17]
	DA0[7:0]	DA0[8]	N/A	N/A	N/A	N/A	QB0[7:0]	QB0[8]
	DB0[16:9]	DB0[17]	N/A	N/A	N/A	N/A	QB0[16:9]	QB0[17]
	DB0[7:0]	DB0[8]	N/A	N/A	N/A	N/A	QB0[7:0]	QB0[8]
256*36(9K 模块 0)	DA1[16:9]	DA1[17]	N/A	N/A	N/A	N/A	QB1[16:9]	QB1[17]
	DA1[7:0]	DA1[8]	N/A	N/A	N/A	N/A	QB1[7:0]	QB1[8]
	DB1[16:9]	DB1[17]	N/A	N/A	N/A	N/A	QB1[16:9]	QB1[17]
	DB1[7:0]	DB1[8]	N/A	N/A	N/A	N/A	QB1[7:0]	QB1[8]

DRM_FIFO 位宽模式

FIFO在丢包重发模式下, wr_eop信号需要占用一位数据位，FIFO数据位宽需要进行如下配置：

表 28 FIFO 丢包重发模式地址深度与数据宽度配置列表

FIFO 地址与位宽配置	FIFO 可用数据位	wr_eop 占用数据位
2K*9	D[7:0]	D[8]
1K*18	D[17:9],D[7:0]	D[8]
512*36	D[35:27],D[25:0]	D[26]

含丢包重发模式的DRM_FIFO支持如表29所示的数据位宽：

表 29 含丢包重发模式的异步/同步 FIFO 模式数据位宽列表

		写端口		
		2K*9	1K*18	512*36
读端口	2K*9	√		
	1K*18		√	
	512*36			√

不含丢包重发模式的DRM_FIFO支持如下表所示的数据位宽：

表 30 不含丢包重发模式的异步/同步 FIFO 模式数据位宽列表

		写端口					
		16K*1	8K*2	4K*4	2K*9 (2K*8)	1K*18 (1K*16)	512*36 (512*32)
读端口	16K*1	√					
	8K*2		√				
	4K*4			√			
	2K*9(2K*8)				√		
	1K*18(1K*16)					√	
	512*36(512*32)						√

免责声明

版权声明

免责声明

1、本文档仅提供阶段性信息，所含内容可根据产品的实际情况随时更新，恕不另行通知。如因本文档使用不当造成的直接或间接损失，本公司不承担任何法律责任。

2、本文档按现状提供，不负任何担保责任，包括对适销性、适用于特定用途或非侵权性的任何担保，和任何提案、规格或样品在他处提到的任何担保。本文档在此未以禁止反言或其他方式授予任何知识产权使用许可，不管是明示许可还是暗示许可。

3、公司保留任何时候在不事先声明的情况下对公司系列产品相关文档的修改权利。